

# (12) UK Patent Application (19) GB (11) 2 366 037 (13) A

(43) Date of A Publication 27.02.2002

(21) Application No 0103233.3

(22) Date of Filing 09.02.2001

(30) Priority Data

(31) 09512560

(32) 24.02.2000

(33) US

(71) Applicant(s)

**International Business Machines Corporation**  
(Incorporated in USA - New York)  
Armonk, New York 10504, United States of America

(72) Inventor(s)

**Yudong Sun**

(74) Agent and/or Address for Service

**J P Richards**  
**IBM United Kingdom Limited, Intellectual Property**  
**Law, Hursley Park, WINCHESTER, Hampshire,**  
**SO21 2JN, United Kingdom**

(51) INT CL<sup>7</sup>

**G06F 17/30**

(52) UK CL (Edition T )

**G4A AUDB**

**H4T TCXX**

(56) Documents Cited

**EP 1063597 A2**

(58) Field of Search

**UK CL (Edition S ) G4A AUDB , H4T TBAX TBBX TBCX**  
**TCGX TCXX**

**INT CL<sup>7</sup> G06F 17/00 17/30**

**Online:WPI, EPODOC, JAPIO, INSPEC**

(54) Abstract Title

**Customising an HTML document**

(57) A request reception module 54 receives a request for a document stored within document server 58. A parsing module 60 parses the document to generate therefrom a corresponding document object model (DOM) 62 including at least one object 64. A style sheet access module 66 obtains a style sheet including at least one rule directed to a target device 50 for displaying the document. A style sheet application module 72 applies the at least one rule of the style sheet to the DOM. A flattening module 76 flattens the DOM to generate therefrom a corresponding transformed document 78. A transmission module 80 transmits the transformed document to a requesting client program.

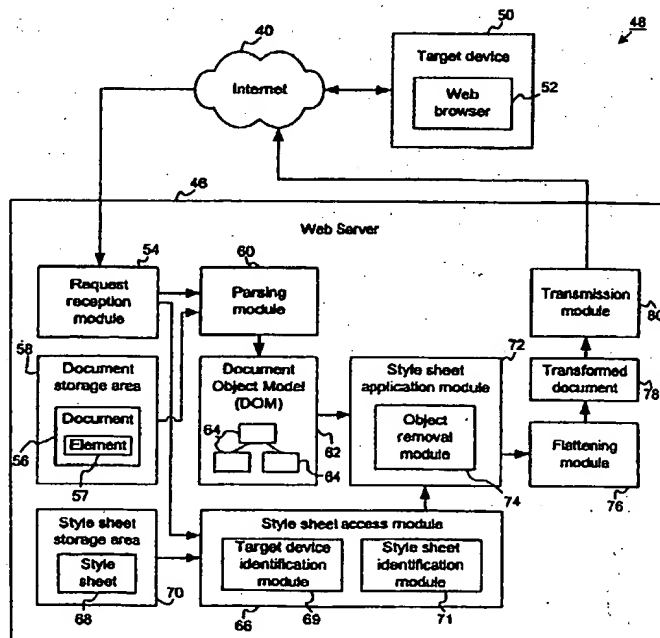


Fig. 2

GB 2 366 037 A

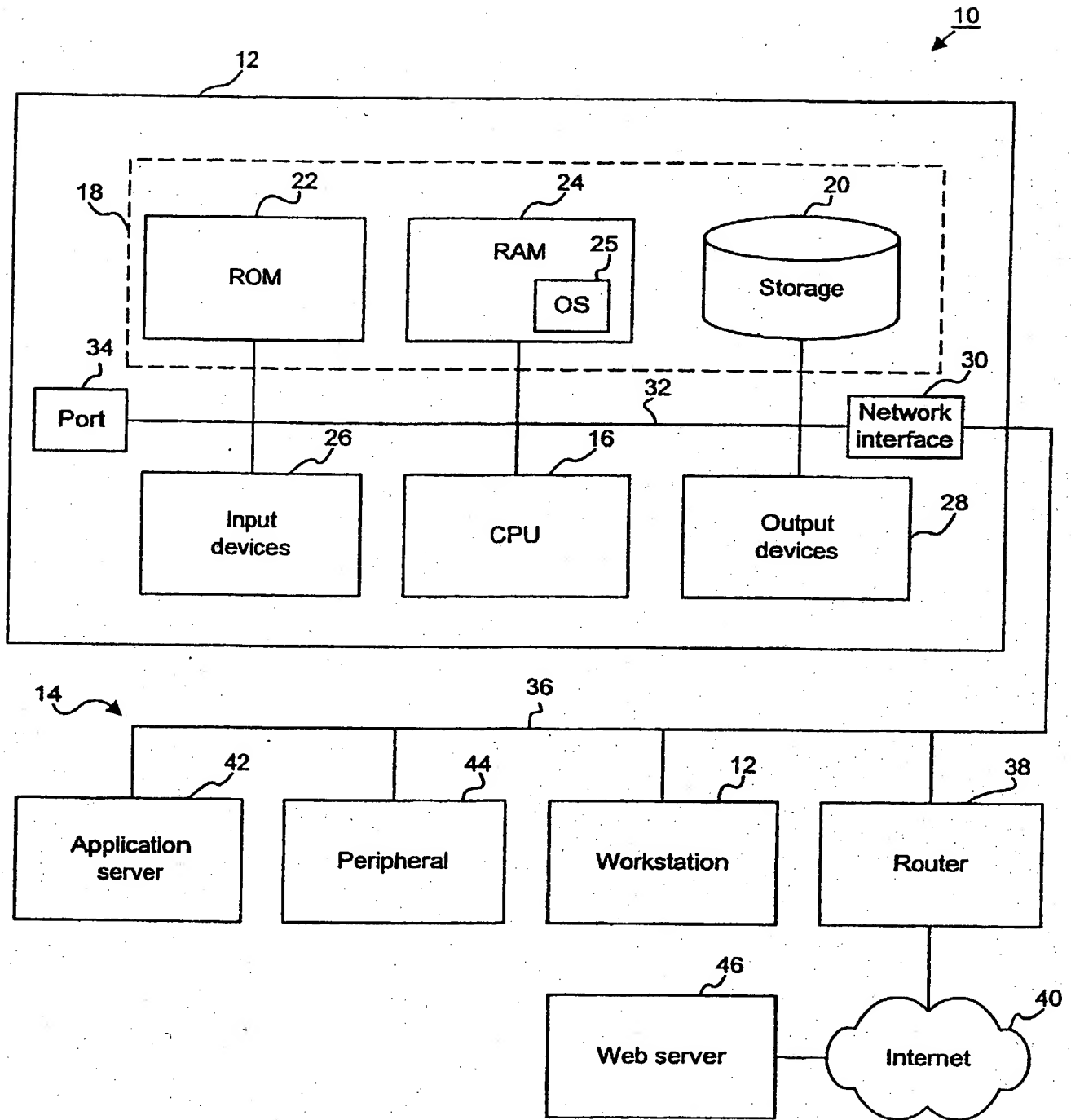


Fig. 1

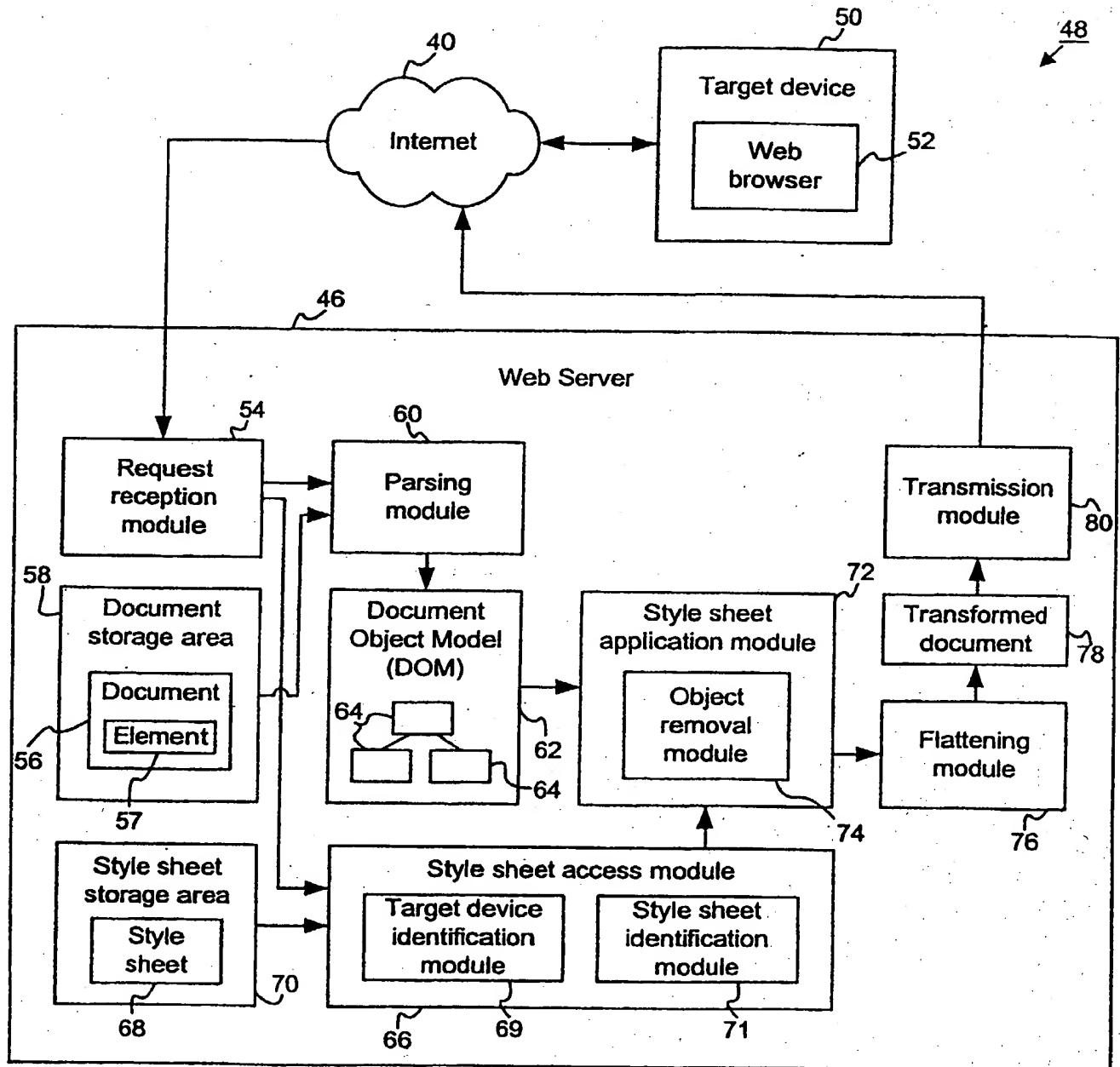


Fig. 2

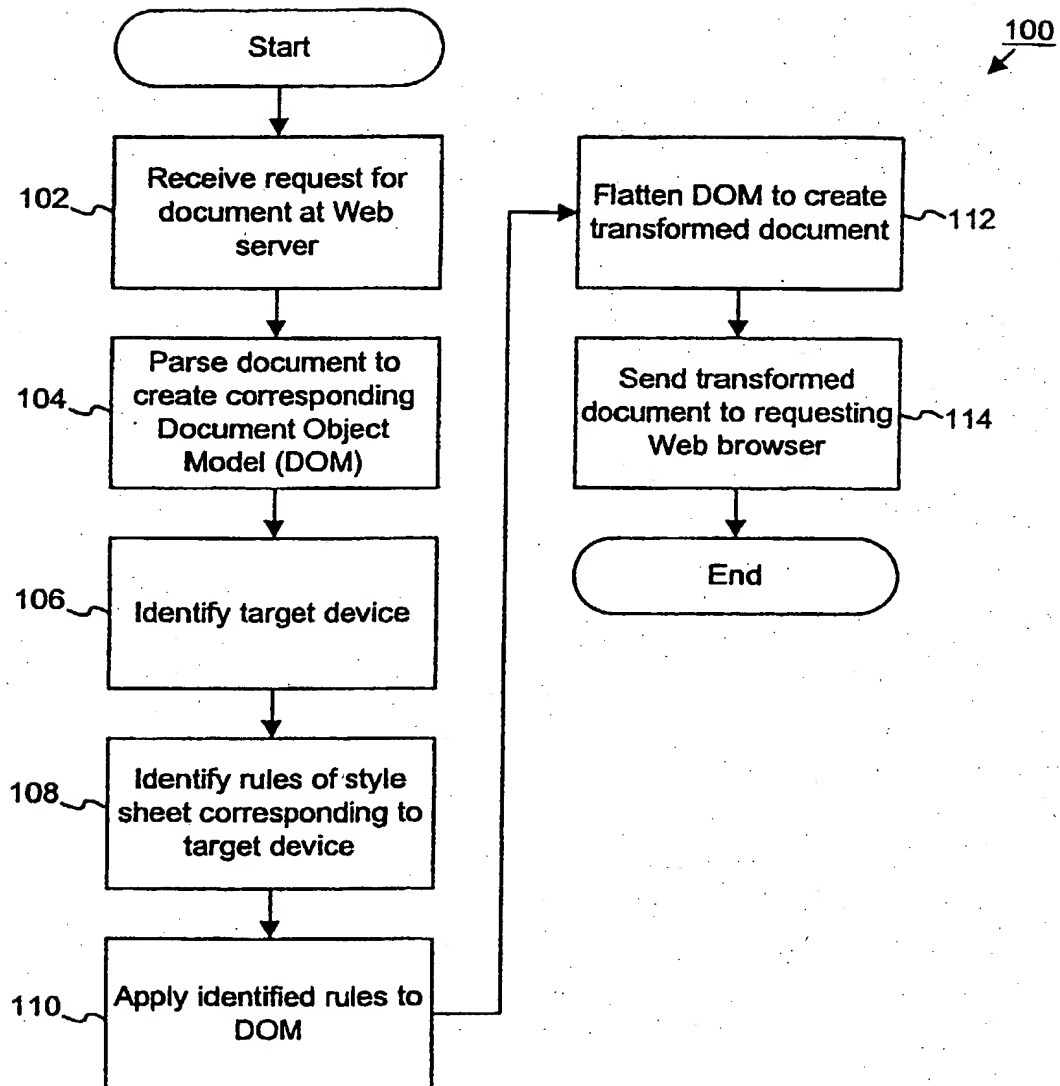


Fig. 3

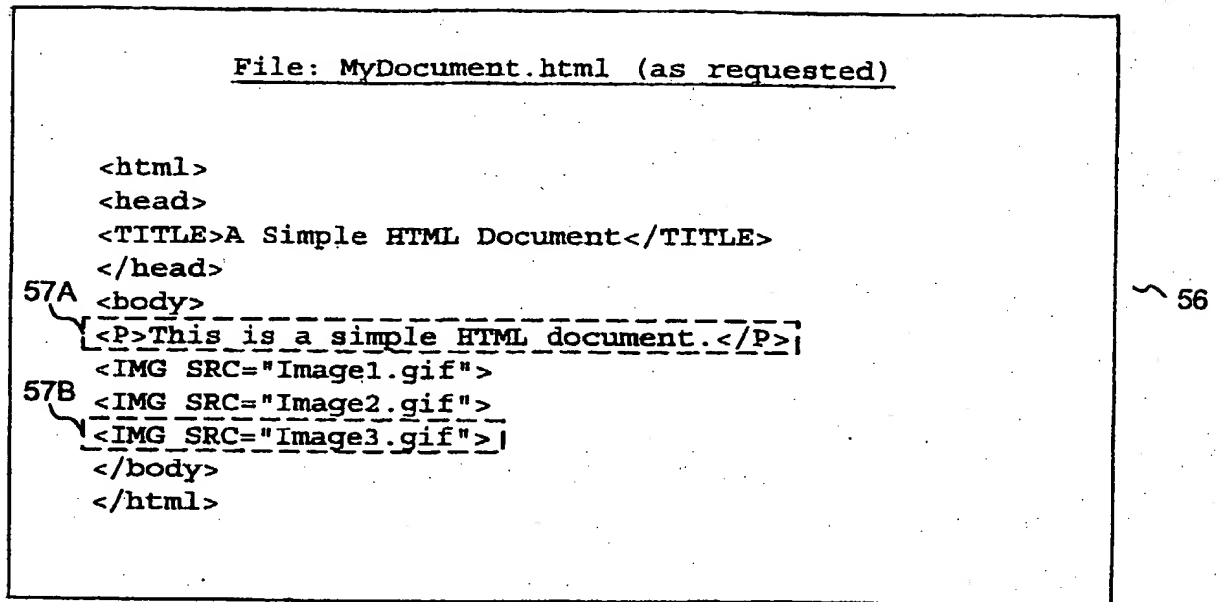


Fig. 4

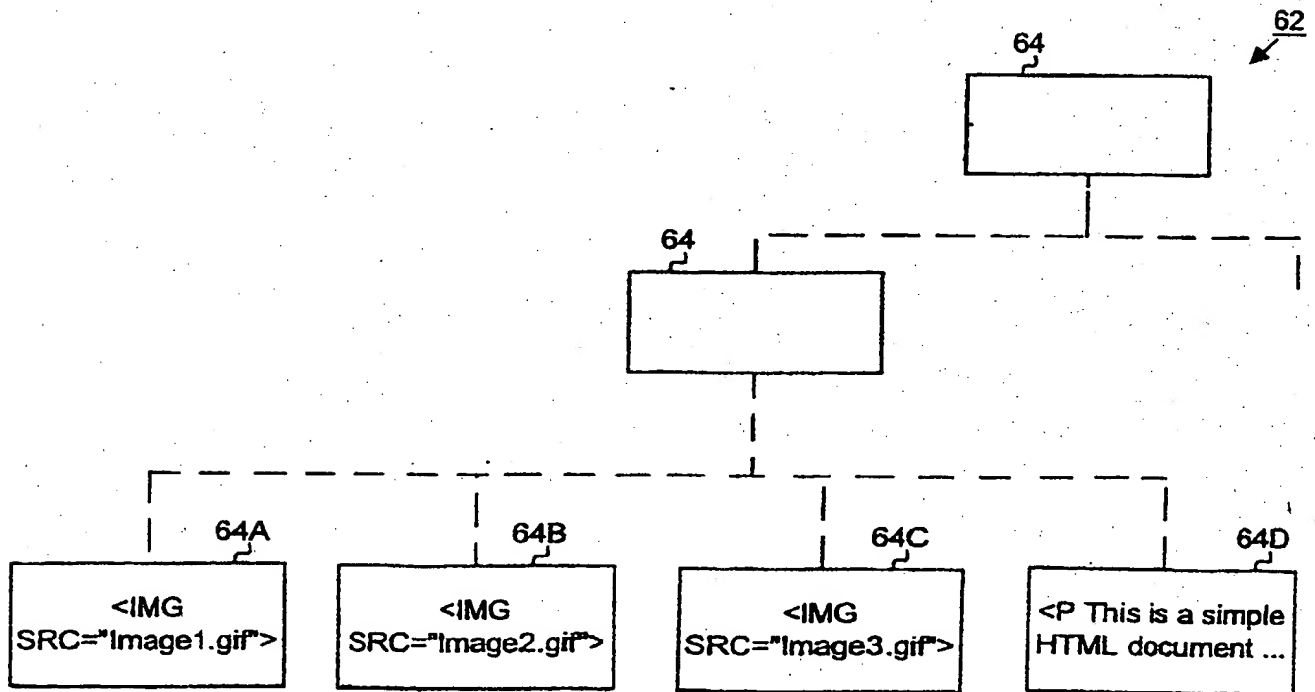


Fig. 5

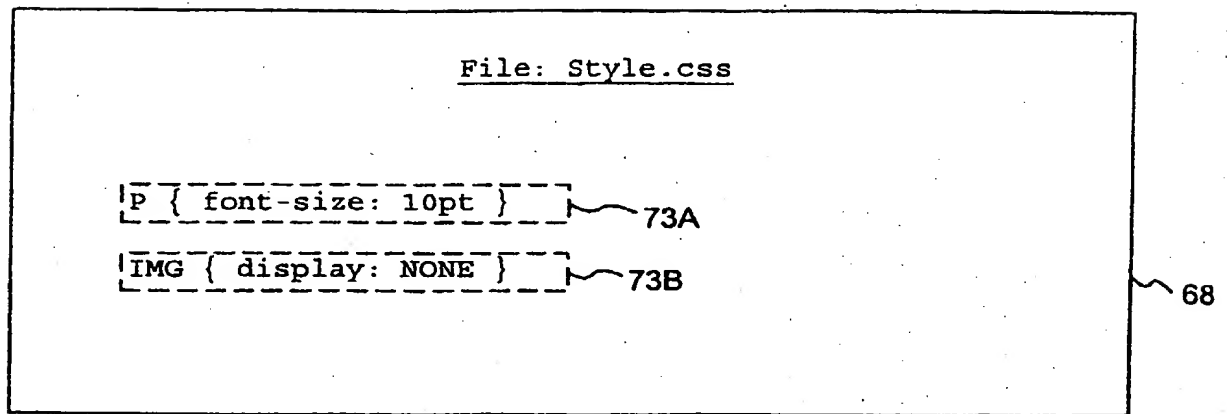


Fig. 6

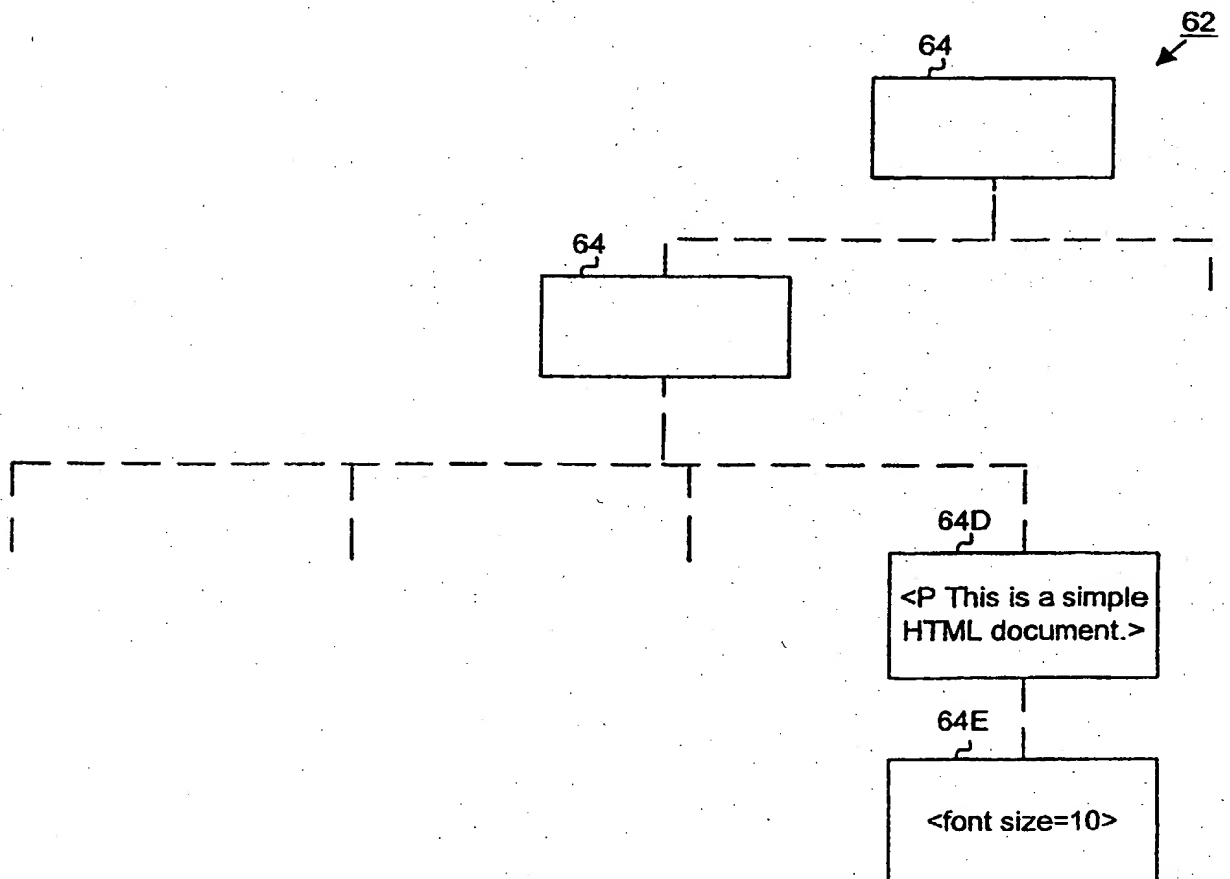


Fig. 7

File: MyDocument.html (as transmitted)

```
<html>
<head>
<TITLE>A Simple HTML Document</TITLE>
</head>
<body>
<P><font_size=10>This is a simple HTML document.</P>
</font>
</body>
</html>
```

78

57C

Fig. 8

## CUSTOMISING AN HTML DOCUMENT

The present invention relates generally to techniques for processing a  
hypertext markup language (HTML) document. More particularly, the present  
invention relates to a system and method for server-side HTML customization  
based on style sheets and a target device.

The World Wide Web (hereinafter "the Web") is a collection of  
Internet-accessible servers from which specially formatted documents may be  
retrieved and displayed by Web browsers, such as Netscape Navigator™ and  
Microsoft Internet Explorer™. Currently, the hypertext markup language  
("HTML") is the most common authoring language for creating Web documents,  
also known as Web pages. A Web page is identified by a uniform resource  
locator ("URL"), which is used by a Web browser to locate and display a  
particular Web page.

Web browsers are now found in a variety of target devices, some of which  
are not capable of displaying every possible Web page. For example, a  
personal data assistant (PDA) is a handheld device that often includes a  
Web browser. However, a PDA is typically limited to displaying a few lines  
of text, and may not be able to display images or other graphical objects.  
As such, specially modified Web pages are typically required for PDAs.

In addition, some target devices have bandwidth limits for accessing the  
Internet. Wireless devices, for instance, such as Web-enabled cellular  
phones, are not capable of rapidly processing large Web pages.  
Accordingly, specially modified versions of Web pages are also desirable  
in the context of limited-bandwidth target devices.

Unfortunately, providing target device-specific versions of Web pages  
usually means providing separate Web pages identified by different URLs,  
which is problematic for a number of reasons. For example, a Web page  
developer would need to create and maintain (e.g. update) several different  
Web pages, resulting in increased costs and the possibility of inconsistent  
versions. Moreover, separate indexes and links would need to be created  
for Web pages corresponding to various target devices, greatly increasing  
the sizes of current indexes and Web pages.

Various techniques have been developed for dynamically customizing a Web  
page for display by different systems. For example, style sheets allow Web  
page developers to define how various HTML elements appear in the context  
of one or more Web pages. An element is a fundamental component of the  
structure of a HTML document, and may include, for example, a table, a  
paragraph, a list, an in-line image, and the like.



Each element may have an associated style, including one or more formatting parameters that dictate how the element is to be displayed by a Web browser. For example, a style may include parameters directed to margins, alignment, color, size, and the like.

Once created, a style sheet may be applied to one or more Web pages. In the case of "cascading" style sheets (CSS), multiple style sheets may be applied to the same Web page. CSS is a well known standard developed by W3C. Currently, CSS is not supported by all Web browsers, although the standard is growing in popularity.

A style sheet may be linked to an HTML document by means of a LINK element:

```
<HEAD>
<LINK REL=STYLESHEET HREF="style.css" TYPE="text/css">
</HEAD>
```

External data files containing style information are typically identified by a ".css" extension, e.g., "style.css."

A style sheet typically includes one or more rules, which define the styles to be applied to various elements or element types before the document is displayed. A rule typically includes at least one selector and at least one style to be attached to that selector. For example, in the rule, P {fontsize: 10pt}, the selector, P, is referred to as a "type" selector, and the style declaration, {fontsize: 10pt}, represents the style to be associated with every HTML element of the type, P (the "paragraph" element).

Style sheets are normally processed on the "client side," i.e. by a Web browser, rather than on the "server side," i.e. by a Web server. The reason for this distinction lies in the fact that Web browsers include parsers, which parse the Web page into a suitable data structure, such as a parse tree. The complex manipulations required for style processing must be performed on a parse tree or the like, and parsing is a normal step in displaying a Web page by a Web browser.

Web servers, on the other hand, do not conventionally parse Web pages, as such is not required to deliver (serve) Web pages. Likewise, Web servers do not normally include parsers. As a result, conventional Web servers are incapable of processing style sheets.

Unfortunately, many Web browsers do not support style sheet processing. For example, a PDA typically has a limited memory and central processing unit (CPU). Accordingly, PDA-based Web browsers are not able to process style sheets. Likewise, many older Web browsers do not support style sheets, since the technology is relatively new and the standards are still in flux.

Accordingly, what is needed is a system and method for server-side HTML customization. This need is met by the invention claimed in claim 1.

An embodiment of the invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 is a schematic block diagram of a computer system suitable for hosting a plurality of software modules which operate according to an embodiment of the invention;

Figure 2 is a schematic block diagram of a system for server-side customization of a hypertext markup language (HTML) document based on style sheets and a target device;

Figure 3 is schematic flowchart of a method for server-side HTML customization based on style sheets and a target device;

Figure 4 is an illustration of an HTML document;

Figure 5 is an illustration of a Document Object Model (DOM);

Figure 6 is an illustration of a style sheet;

Figure 7 is an illustration of a transformed DOM: and

Figure 8 is an illustration of a transformed HTML document;

Throughout the following description, various system components are referred to as "modules." In certain embodiments, the modules may be implemented as software, hardware, firmware, or any combination thereof.

For example, as used herein, a module may include any type of computer instruction or computer executable code located within a memory device and/or transmitted as electronic signals over a system bus or network. An identified module may include, for instance, one or more physical or logical blocks of computer instructions, which may be embodied within one or more objects, procedures, functions, or the like.

The identified modules need not be located physically together, but may include disparate instructions stored at different memory locations, which together implement the described logical functionality of the module. Indeed, a module may include a single instruction, or many instructions, and may even be distributed among several discrete code segments, within different programs, and across several memory devices.

Figure 1 is a schematic block diagram of a computer system 10 in which a plurality of software modules may be hosted on one or more computer workstations 12 connected via a network 14. The network 14 may include a wide area network (WAN) or local area network (LAN) and may also include an interconnected system of networks, one particular example of which is the Internet.

A typical computer workstation 12 may include a central processing unit (CPU) 16. The CPU 16 may be operably connected to one or more memory devices 18. The memory devices 18 are depicted as including a non-volatile storage device 20 (such as a hard disk drive or CD-ROM drive), a read-only memory (ROM) 22, and a random access memory (RAM) 24.

The computer workstation 12 may operate under the control of an operating system (OS) 25, such as OS/2®, WINDOWS NT®, WINDOWS®, UNIX®, and the like. In various embodiments, the OS 25 provides a graphical user interface (GUI).

The computer workstation 12 may also include one or more input devices 26, such as a mouse and/or a keyboard, for receiving inputs from a user. Similarly, one or more output devices 28, such as a monitor and/or a printer, may be provided within, or be accessible from, the computer workstation 12.

A network interface 30, such as an Ethernet adapter, may be provided for coupling the computer workstation 12 to the network 14. Where the network 14 is remote from the computer workstation 12, the network interface 30 may include a modem, and may connect to the network 14 through a local access line, such as a telephone line.

Within any given computer workstation 12, a system bus 32 may operably interconnect the CPU 16, the memory devices 18, the input devices 26, the output devices 28, the network interface 30, and one or more additional ports 34, such as parallel and/or serial ports.

The system bus 32 and a network backbone 36 may be regarded as data carriers. Accordingly, the system bus 32 and the network backbone 36 may be embodied in numerous configurations, such as wire and/or fiber optic lines, as well as electromagnetic communication channels using visible light, infrared, and radio frequencies.

The computer workstations 12 may be coupled via the network 14 to one or more application servers 42, and/or other resources or peripherals 44, such as scanners, fax machines, and the like. External networks, such as the

Internet 40, may be coupled to the network 14 through a router 38 or firewall.

5 In various embodiments, one or more Web servers 46 may be accessible to the workstations 12 via the Internet 40. A Web server 46 may be implemented using a workstation 12, as described above, including specialized software for delivering (serving) Web pages to Web browsers. A variety of Web server application programs are available, including public domain software from the National Center for Supercomputing Applications (NCSA) and Apache, 10 as well as commercial packages from Microsoft, Netscape and others.

Referring now to Figure 2, a system 48 for server-side HTML customization may include a Web server 46 and a target device 50. The target device 50 may be implemented using a workstation 12, which includes a Web browser 52, 15 such as Netscape Navigator™ or Microsoft Internet Explorer™. The Web browser 52 may be configured to communicate with the Web server 46 via the hypertext transfer protocol ("HTTP").

20 In various embodiments, the target device 50 may include a standard desktop computer, such as an IBM PC™ or compatible. In alternative embodiments, however, the target device 50 may include a Web-enabled personal data assistant (PDA), such as a PalmPilot™ VII, available from 3Com Corporation, or the like.

25 The Web server 46 is depicted as including a request reception module 54. In one embodiment, the request reception module 54 receives (from the Web browser 52) a request for a document 56 stored within a document storage area 58 of the Web server 46. The document 56 may be encoded in the hypertext markup language ("HTML") and may include one or more HTML 30 elements 57, as described more fully hereafter.

In one embodiment, the Web server 46 also includes a parsing module 60, commonly referred to as a "parser." The parsing module 60 retrieves, in various embodiments, the requested document 56 and parses the document 56 35 to generate therefrom a corresponding Document Object Model (DOM) 62, often referred to as a "parse tree." A DOM 62 is a tree-like, hierarchical data structure including one or more objects 64 that represent the various HTML elements 57 of the document 56.

40 In certain embodiments, the parsing module 60 is a conventional HTML parser. For example, both Netscape Navigator™ and Microsoft Internet Explorer™ include HTML parsers, which may be adapted, in various embodiments, for use within the Web server 46. In an alternative embodiment, a custom HTML parser may be used. Conventionally, however, a

Web server 46 does not include a parsing module 60, since a document 56 is normally parsed only by a Web browser 52 at the time the document 56 is displayed.

5 The Web server 46 may also include a style sheet access module 66. In certain embodiments, the style sheet access module 66 is configured to retrieve a style sheet 68 (from a style sheet storage area 70) including one or more rules directed to a target device 50.

10 The style sheet access module 66 may include a target device identification module 69, which may identify the type or class of the target device 50. This may be accomplished, for example, based on platform information provided as part of a browser request. Typically, a browser request includes a browser name and version, as well as information about the platform, such as screen resolution.

15 The style sheet access module 66 may also include a style sheet identification module 71. According to various embodiment, a single style sheet 68 may include rules directed to different target devices 50. For example, rules directed to a PDA-type device may be identified within the style sheet 68 by @media handheld indicator or the like. Consequently, the style sheet identification module 71 may identify the rules of the style sheet 68 corresponding to the identified target device 50.

20 The Web server 46 may also include a style sheet application module 72, which applies the appropriate rules of the style sheet 68 to the DOM 62 of the document 56. Techniques for applying style sheets rules are well known in the art. For example, both Netscape Navigator™ and Microsoft Internet Explorer™ include style sheet application modules 72, which may be adapted, in various embodiments, for use within the Web server 46. In an alternative embodiment, however, a custom style sheet application module 72 may be used.

25 In one embodiment, the style sheet access module 66 includes an object removal module 74. Where, for instance, a rule within a style sheet 68 indicates a "NONE" display style, or similar designation, for an element 57 or element type, a corresponding object 64 within the DOM 62 is preferably removed.

30 For example, the rule, IMG { display: NONE }, indicates a "NONE" display style for the IMG (in-line image) element type. Accordingly, the object removal module 74 preferably removes the object(s) 64 of the DOM 62 corresponding to in-line image elements 57. This is advantageous, for

instance, where a document 56 includes in-line images, but a target device 50, such as a PDA, cannot display such images.

5 The style sheets 68 and the Web documents 56 are depicted as logically separate data files, and may even be stored within separate storage areas 58, 70 of the Web server 46. In an alternative embodiment, a style sheet 68 may be included within a separate portion of the document 56. For example, the HTML elements 57 of the document 56 and the rules of the style sheet 68 may be stored within separate portions of a single logical data  
10 file.

The Web server 46 may also include a flattening module 76. In various embodiments, the flattening module 76 flattens the DOM 62 to generate therefrom a corresponding transformed document 78. As used herein, the  
15 term "flattening" refers to a process of converting the DOM 62 back into an equivalent HTML document 86 including one or more corresponding HTML elements 57. Techniques for flattening a DOM 62 are well known in the art.

20 The resulting document 86 is designated as "transformed" because the style sheet application will be reflected in the HTML elements 57 of the transformed document 78.

In various embodiments, the Web server 46 may also include a transmission module 80. The transmission module 80 may send the transformed document  
25 78 (via the Internet 40) to the Workstation 12, such that the document 86 may be displayed by the Web browser 52.

Referring now to Figure 3, a schematic flowchart includes a method 100 for server-side HTML customization according to a presently preferred  
30 embodiment of the invention. The method 100 may begin by receiving 102, at a Web server 46, a request for a document 56.

Figure 4 illustrates an exemplary document 56 according to an embodiment of the invention. The document 56 may include one or more HTML elements 57,  
35 such as a paragraph element 57A and an image element 57B.

After the document request is received 102, the method 100 may continue by parsing 104 the document 56 to generate therefrom a corresponding Document Object Model (DOM) 62. As noted, a DOM 62 is a tree-like, hierarchical  
40 data structure including one or more objects 64 that represent the HTML elements 57 of the document 56. Figure 5 illustrates a portion of a simplified DOM 62 corresponding to the document 56 of Figure 4.

After the document 56 is parsed 104, the method 100 may continue by identifying 106 a target device 50 for displaying the document 56. As noted, the target device 50 may be based on platform information provided by a browser request.

After the target device 50 is identified 106, the method 100 may continue by identifying 108 one or more rules of a style sheet 68 directed to the identified target device 50. As noted, a single style sheet 68 may include sets of rules directed to different target devices 50. For example, a rule set directed to a PDA-type device may be identified by a @media handheld indicator or the like. Consequently, the style sheet identification module 71 may identify the rules of the style sheet 68 directed to the identified target device 50.

Figure 6 illustrates an exemplary style sheet 68 for a PDA-type target device 50 according to an embodiment of the invention. The style sheet 68 may include any number of standard rules 72, such as rule-sets and at-rules (as defined in the CSS standard).

As previously explained, a PDA may not be capable of displaying images or other graphical objects. In addition, a PDA may be limited as to fonts, font sizes, and the like. Moreover, limited-bandwidth target devices 50, such as wireless devices, may require Web documents 56 that have reduced graphical content. The style sheet 68 may include one or more rules 72 for customizing a Web document 56 for a target device.

For example, a first rule 73A, i.e. `P { font-size: 10pt }` may set the font size for each paragraph element 57. Specifically, the rule 73A may set the font size to 10 points.

A second rule 73B, i.e. `IMG { display: NONE }`, may not include a typical style declaration, but may specify "NONE" display style or a similar designation. In various embodiments, a "NONE" display style causes the object removal module 74 to remove objects 64 corresponding to the element type specified in the rule 73.

After the style sheet 68 is identified, the method 100 may continue by applying 110 the identified style sheet rules 73 to the DOM 62. Each rule 73 of the style sheet 68 may be applied to the objects 64 of the DOM 62, which may result in the removal of certain objects 64 and the addition of others.

For example, as illustrated in Figure 7, the rule 73A may add a new object 64E, corresponding to a `<font size=10>` element 57. By contrast, the rule

73B may cause objects 64A-C (IMG elements 57) of Figure 5 to be deleted. After application of the style sheet 68, the DOM 62 may appear as shown in Figure 7.

While the style sheet 68 and the document 56 are depicted herein as logically separate data files, the style sheet 68 may be included, in some instances, within a separate portion of document 56. For example, all of the rules 73 of the style sheet 68 may be located, as a group, at the beginning of the document 56:

```

<style>
  P { font-size: 10pt }
  IMG { display: NONE }
</style>
<html>
<head>
<TITLE>A Simple HTML Document</TITLE>
</head>
<body>

```

In alternative embodiments, a single style sheet 68 may include portions corresponding to two or more target devices 50. For example, a style sheet 68 may include the following:

```

@media handheld
{
  P { font-size: 10pt }
  IMG { display: NONE }
}
@media tinyscreen
{
  P { font-size: 12pt }
  IMG { display: NONE }
}

```

In such an embodiment, the style sheet access module 66 may parse the style sheet 68 and extract the rules 73 corresponding to the identified target device 50.

After the rules 73 have been applied 110, the method 100 may continue by flattening 112 the DOM 62 to create a transformed document 78, which may then be sent 114 to the requesting Web browser 52 for display. As previously noted, the flattening process involves converting the DOM 62



back into an HTML document 86. Consequently, any transformations to the DOM objects 64 will be preferably reflected in the corresponding HTML elements 57 of the document 86.

For example, Figure 8 illustrates an exemplary transformed document 78 after flattening 116 the DOM 62 of Figure 7. Comparing the transformed document 78 of Figure 8 to the requested document 56 of Figure 4 reveals that a new HTML element 57C is added, and the image elements 57 of Figure 4, including element 57B, are deleted.

Based on the foregoing, the present invention offers a number of advantages not found in conventional approaches. Style sheets 68 are processed on the server side, which is advantageous for target device 50 that are not capable of style sheet processing, such as PDAs.

Moreover, the system and method of the present invention make it possible to maintain one version of a Web document 56 for a variety of target devices 50, each of which may have different capabilities. Thus, different target devices 50 may access a Web document 56 using the same URL, which minimizes development and maintenance costs and the need for multiple links for different target devices 50.

Even target devices 50 that are capable of processing style sheets 68 may benefit from the present invention, such as those with a limited bandwidth (e.g. wireless devices). Because style sheets 68 are conventionally applied by a Web browser 52, a wireless target device 50 must first retrieve a document 56 and a corresponding style sheet 68 before the style sheet 68 may be applied. Unfortunately, if the document 56 is large, the bandwidth has already been wasted.

By contrast, the system and method of the present invention apply style sheets 68 on the Web server 46. Server-side HTML customization results in a more compact document 56 that may be sent to a target device 50 over a limited-bandwidth network. Moreover, the need for bandwidth is further reduced because the style sheets 68 are never sent to the target device 50.

CLAIMS

1. A computer-implemented method for customizing a requested document comprising at least one hypertext markup language (HTML) element, the method comprising:
  - parsing the document to generate therefrom a corresponding document object model (DOM) including at least one object;
  - obtaining a style sheet including at least one rule directed to a target device;
  - applying the at least one rule of the style sheet to the DOM; and
  - flattening the DOM to generate therefrom a corresponding transformed document suitable for display by the target device.
2. The method of claim 1, wherein the style sheet comprises a cascading style sheet (CSS).
3. The method of claim 1, wherein the obtaining step comprises:
  - identifying a target device for displaying the document; and
  - identifying at least one rule of a style sheet directed to the identified target device.
4. The method of claim 3, further comprising:
  - receiving a request for a document from a client program.
5. The method of claim 4, wherein the client program comprises a Web browser.
6. The method of claim 1, wherein the style sheet includes rules directed to at least two different target devices.
7. The method of claim 1, wherein the style sheet is stored within a separate portion of the document.
8. The method of claim 1, wherein the style sheet and the document are stored as logically separate data files.
9. The method of claim 1, further comprising:
  - transmitting the transformed document to a client program.
10. The method of claim 1, the transforming step comprising:

removing at least one object of the DOM in response to an indication within the style sheet to remove a corresponding HTML element from the document.

5 11. A system for customizing a requested document comprising at least one hypertext markup language (HTML) element, the system comprising:  
means for parsing the document to generate therefrom a corresponding document object model (DOM) including at least one object;

10 means for obtaining a style sheet including at least one rule directed to a target device;

means for applying the at least one rule of the style sheet to the DOM; and

means for flattening the DOM to generate therefrom a corresponding transformed document suitable for display by the target device.

15 12. An article of manufacture comprising a computer readable program storage medium bearing instructions executable by the computer to perform the method claimed in any one of claims 1 - 10.



INVESTOR IN PEOPLE

Application No: GB 0103233.3  
Claims searched: 1 to 12

Examiner: John Donaldson  
Date of search: 14 December 2001

## Patents Act 1977 Search Report under Section 17

### Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK CI (Ed.S): G4A(AUDB); H4T(TBAX, TBBX, TBCX, TCGX, TCXX)

Int CI (Ed.7): G06F 17/00, 17/30

Other: Online:WPI, EPODOC, JAPIO, INSPEC

### Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A, P	EP 1063597 A2 (SONY), see abstract	-

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.